**TEST PLAN**

**Project Name: Hunt the Wumpus**

**Document date:**

Table of Contents

Introduction3

1.1 [Test Environment & Tools](https://docs.google.com/document/d/1wbyw7A4x3RF5L--A0d9J9Viu4Qg_gvM1W7209LNB3a8/edit#heading=h.3znysh7) 3

[1.2 Testing Team](https://docs.google.com/document/d/1KT4hPDs1Z9B5_OKNuDG6DTPrdrieuDNg/edit#heading=h.tyjcwt) 3

1.3 QA Role and Responsibilities 4

1.4 Reference documents 4

1.5 In Scope 4

1.6 Out of Scope 4

1.7 Test Strategy 5

2.1 Execution Strategy 5

2.2 [Test Levels](https://docs.google.com/document/d/1KT4hPDs1Z9B5_OKNuDG6DTPrdrieuDNg/edit#heading=h.3dy6vkm) 5,6,7

2.3 Regression testing 7

2.4 Integration testing 7

2.5 Performance testing 8

2.6 Endurance testing 8

2.7 Security testing 8

2.8 API testing 8

3.1 Test Results 9

3.2 Estimations 9

3.3 Test metrics 9

3.4 Test Risk 9

# 

# **Introduction**

This test plan describes the testing approach and overall framework that will drive the testing of the online game Hunt the Wumpus by applying Functional and Non - functional testing

| Product/Feature | Testing Period | Tested Builds | Deployment Date |
| --- | --- | --- | --- |
| Online Game: Hunt the Wumpus |  |  |  |

## Test Environment

| Test Environment | Software Details | Comments |
| --- | --- | --- |
| Life | Native console  DOT.net 6  Windows 10 desktop version |  |

## Tools: YouTrack, Postman

## 1.2 Testing Team

| Role | Name | Comments (e.g. test effort h) |
| --- | --- | --- |
| Junior QA | Borbely Adela  Paula Nan  Emese Egeto |  |

# 1.3 QA role and responsibilities

Develop test scenarios, test steps, and test expected results

Interact with the application

Identity, document and prioritize defects

Perform execution and validation

1.4 Reference Documents:

- Software requirements Specifications:

https://github.com/BackToTech-Study/HuntTheTwistedWumpus https://github.com/BackToTech-Study/HuntTheTwistedWumpus/tree/main/GameServer/Resources

## 1.5 In Scope

The test plan includes the Functional and Non-Functional testing performed on online game Hunt the Wumpus.

The software test will ensure that our software is free from input errors and all the game logic will function as intended based on requirements.

## 1.6 Out of Scope

All features except that is mentioned under In Scope

1.7 Test Strategy

We need to perform Functional testing and Non - Functional testing, for the functionality which is mentioned in the above Scope.

As part of Functional Testing and Non -Functional Testing, we are going to follow the below approach for testing this project

Step 1 - Creation of Test Scenarios for the functionality that are in Scope

We will use our expertise in creating the Test Scenarios by applying the Functional Testing and Non Functional Testing

Step 2 - Testing the functionality

We will perform manual testing and will report the testing results in YouTrack

**2.1 Execution Strategy**

## Entry and Exit Criteria

1. **Entry criteria**

* all necessary resources are available - test tools are available, test cases are defined, test artifacts are ready and ticket is in the testing state

1. **Exit criteria**

* test scenarios are created and executed
* the rate of tests scenarios passed is satisfactory
* no critical defects were found

2.2 Test levels

Functional testing

* Navigation:

-direct link navigation

- initialization of the game

- cave rooms

- sounds

- actors: players, Wumpus, Giant Bat

- player actions

- Wumpus actions

- Giant Bats actions

-items: pebble/stone, magical Bow, arrow

- Integration Testing

- modules integration

- data communication

- characters interactions

- Exploratory testing

- Regression Testing

-Black box testing techniques

Non-Functional Testing:

* Performance Testing:

- checking the response time

- load testing

- Endurance testing:

- check for memory leaks

- test database connection

* Security testing:

- validity of certification

- no access to the database

- API testing

- verify the status codes - request and response

## 2.3 Regression testing

Regression testing is a type of software testing which ensures that previously developed and tested software application working in the same way as it was working before recent code/configuration changes done. Test cases are re-executed in order to check whether the earlier functionality of the software is working as projected and the new configuration/code changes have not introduced any new issues/bugs. There is a need for regression testing whenever the code is changed, when a new feature is added to the software application, and when a functional or performance defect/issue is fixed. During the regression test phase no changes must be allowed to code, database used for regression testing must be isolated, and no database changes must be allowed, also code being regression tested should be under a configuration management tool.

# 2.4 Integration testing

Integration testing is a type of testing where software modules are integrated logically and tested as a group. The purpose of this level of testing is to expose defects in the interaction between the software modules when they are integrated, it is necessary to verify the software modules work in unity.

# 

# 2.5 Performance testing

Performance testing is a software testing process used for testing the speed, response time, stability, reliability, scalability and resource usage of a software application under a particular workload. We will perform Load testing in order to check the application ability to perform under anticipated user loads. The objective is to identify and eliminate the performance bottlenecks in the software application.

Load testing should be planed once the application becomes functionally stable.

# 2.6 Endurance testing

Endurance testing is a type of testing where software is tested with high load extended over a significant amount of time to evaluate the behavior of software application under sustained use. The purpose is to ensure that the application is capable enough to handle extended load without deterioration of response time.

Endurance testing will be performed at the last stage of the performance run cycle.

# 2.7 Security testing

Security testing is a type of testing that uncovers vulnerabilities, threats, risks in a software application, and prevents malicious attacks from intruders. It is necessary to involve security testing in the SDLC life cycle in the earlier phases.

2.8 API testing

API testing is a type of testing that analyzes an application program interface (API) to verify it fulfills its expected functionality. API(Application Programming interface) is a computing interface that enables communication and data exchange between two separate software systems.

QA team will perform API testing after the build is ready and the tool they will use is Postman.

# 3.1 Test Results

## Executed Test Scenarios

During the testing process, no defects were found.

## 3.2 Estimation

Estimations of the product backlog will be done using cards as user stories, features. Each card should include a brief description of the item, the anticipated completion period, and who its owner is. We prioritize Kanban cards and then organize them based on priority. The highest priority items are the highest on the board, and the lowest is the lowest.

## 3.3 Test Metrics

Test Metrics are used to measure the progress and level of success of the test.

Test preparation and execution were reported to the members of the team and test scenarios were executed 100%.

3.4 Test risk

Testing scope delay due to a new defect found.

Delay in testing schedule due to illness or natural disasters

Testing is done based on incomplete validation, failing to establish testing priorities, or improper communication establishes an operational risk.

Testing was done without access to document design establishes an operational risk

Testing done without access to requirements establishes an operational risk